

# **Catching the Right, Objective Word - Write Up**

Cheng Guo

## **Overview**

As third-year undergraduate students, students of my age must start looking for internships. The process of finding a job both enlarges our vision and deepens our understanding of the industry. When we apply for internships, we are required to present our resumé. During my study in the class HCDE 321: Professional Portfolio, we are encouraged to use strong words in our resumé to keep recruiters hooked as we make our points.

However, there are difficulties, especially for non-native English speakers like me, to find a strong word that can keep recruiters hooked. We may not know that there are strong alternatives to the word we wrote, or we may not know that some words can be used interchangeably to strengthen our statement. For my project, I would like to indicate among all the synonyms of a word, which one is the strongest to use.

## **Proposal**

I propose to build a website that searches for all the synonyms of a given word that the user types, then it will also decide which word is the strongest, and which word has the closest meaning to the given word. My website will benefit all students preparing their resumé because they can find the most suitable words to keep recruiters hooked. It can also be used for other occasions that need to make strong statements, like preparing a speech draft for public speaking.

I will prompt users to type a word they would like to find a stronger replacement. Then, I will present all the synonyms of the given word in a word cloud, where the size of each word is based on how strong they are. A word cloud makes users spot the strongest word easily, and it also provides users with other options so the user can decide which word to use. Creating such a word cloud makes the word-searching process easier when writing the resumé, cover letter, or speech draft.

Based on the user input, I will query the given word in [the Merriam-Webster's Collegiate® Thesaurus API](#) to get a list of synonyms of the given word. Then, I will get the synonyms of all the synonyms to have a large list of words to work

with. Since lists allow duplicates, based on each word's occurrence in the list, I will create the word cloud. The words that appear more times in the big list will have a bigger font since they have a closer meaning to the given word. To determine how strong a word is, I have to consider how many synonyms it has. As I tested several words, a stronger word tends to have more synonyms than a weaker word because a stronger word contains more information than a weaker word. For example, once I learned that "assert" is a stronger alternative to the word "speak," I found that "assert" has more synonyms than "speak." I will also give the stronger words a bigger font. Based on those two factors (how many times they appeared in the big list of synonyms and synonyms of synonyms, and how many synonyms they have), each word will have a different size in the word cloud, and the user can easily spot which word is a better choice.

## **Resource Writeup**

### **[Merriam-Webster's Collegiate® Thesaurus API](#)**

To access the data of this API, the developer needs to register for a key. The developer must provide occupation, application description, and potential launch date to qualify for a key. The Thesaurus API does not contain methods, so the only way to retrieve information is to select a certain word and all the information of this word in JSON format, including synonyms, related words, antonyms, and short definitions. I have attached an example output of the word "python" from the Thesaurus API at the end of this document.

To analyze the data retrieved from the Thesaurus API, I used the approach learned in the class through the urllib library of python. In specific, I used "urllib.parse" to create the URL, "urllib.request" to retrieve information, and "urllib.error" to detect any errors. More information on the urllib library can be accessed on its [website](#).

In my project, since I am also creating word clouds, I used several python libraries to make word clouds. The first one I used is a library called "wordcloud," which is an algorithm dedicated to creating word clouds in python. This library allowed me to create a word cloud based on word frequency, and I can also personalize the word cloud to a specific shape or use various colors. More information about the "wordcloud" library can be found on its [website](#).

Alongside the “wordcloud” library, I also used the “PIL” library to plot a word cloud into an image for showing on the website. The “PIL” is a widely-used library for image processing in python. It can also create an image from code. In specific, I am using the “Image” module in this library to plot. More information of the “Image” module can be found on its [website](#).